OIPE
MAY 2 9 2007
IAP7
PATENT & TRADEMARK OFFICE

TC AF

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200208014-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s):	Christina Hsu, et al.	Confirmation No.:	7237
Application No.:	10/677,004	Examiner:	Pham, Thai V.
Filing Date:	October 1, 2003	Group Art Unit:	2192

Title: METHOD AND APPARATUS FOR SUPPORTING CONFIGURATION OF A WEB APPLICATION IN A WEB PRESENTATION ARCHITECTURE

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on March 26, 2007.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

☐ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

☐ 1st Month $120 ☐ 2nd Month $450 ☐ 3rd Month $1020 ☐ 4th Month $1590

☐ The extension fee has already been filed in this application.

☒ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of $ 500 . At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

☒ A duplicate copy of this transmittal letter is enclosed.

☒ I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450

Date of Deposit: May 23, 2007

OR

☐ I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571)273-8300.

Date of facsimile:

Typed Name: Helen Tinsley

Signature: Helen Tinsley

Respectfully submitted,

Christina Hsu, et al.

By Barry D. Blount

Barry D. Blount

Attorney/Agent for Applicant(s)

Reg No. : 35,069

Date : May 23, 2007

Telephone : 281-970-4545

Rev 10/06a (AplBrief)

OIPE IAP74
MAY 29 2007
PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Christina Hsu et al.	§	Group Art Unit:	2192
Serial No.: 10/677,004	§	Examiner:	Pham, Thai V.
Filed: October 1, 2003	§		
For: Method and Apparatus for Supporting Configuration of A Web Application in A Web Presentation Architecture	§	Atty. Docket:	200208014-1 NUHP:0121/FLE/BLT

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING
37 C.F.R. 1.8

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date below:

May 23, 2007 — Helen Tinsley
Date — Helen Tinsley

APPEAL BRIEF PURSUANT TO 37 C.F.R. §§ 41.31 AND 41.37

This Appeal Brief is being filed in furtherance to the Notice of Appeal mailed on March 21, 2007, and received by the Patent Office on March 26, 2007.

1. REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Development Company, L.P., the Assignee of the above-referenced application by virtue of the Assignment to Hewlett-Packard Development Company, LP, recorded at reel 014581, frame 0608, on October 1, 2003. Accordingly, Hewlett-Packard Development Company, L.P., will be directly affected by the Board's decision in the pending appeal.

05/30/2007 MGEBREM1 00000037 082025 10677004
01 FC:1402 500.00 DA

2. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any other appeals or interferences related to this Appeal. The undersigned is Appellants' legal representative in this Appeal.

3. STATUS OF CLAIMS

Claims 1-24 are currently pending, are currently under final rejection and, thus, are the subject of this Appeal.

4. STATUS OF AMENDMENTS

As the instant claims have not been amended at any time, there are no outstanding amendments to be considered by the Board.

5. SUMMARY OF CLAIMED SUBJECT MATTER

The present invention relates generally to the field of web applications. Particularly, the present invention relates to a controller generator adapted to provide a web application with a controller for receiving a request from a user, and responding by sending information to the user. The present invention further relates to a configurator generator adapted to provide configurator that loads configuration in information for the controller and stores the configuration for subsequent access. *See* Abstract.

The Application contains four independent claims, namely, claims 1, 8, 15 and 22 all of which are the subject of this Appeal. The subject matter of these claims is summarized below.

With regard to the aspect of the invention set forth in independent claim 1, discussions of the recited features can be found at least in the below-cited locations of the specification and drawings. By way of example, an embodiment in accordance with claim 1 provides a "system for creating web applications." The system of claim 1 comprises "a controller generator (e.g., 102) that is adapted to provide a web application (*e.g.*, 204) with a controller (*e.g.*, 18, 208) that receives a request (e.g., 148) for data from a user and responds to the request by sending information to the user...." *See, e.g., id.* at paragraphs 10, 15,18, 19, 21, 22 and 36; *see also* FIGS. 1-3. Claim 1 further recites "a configurator generator that is adapted to provide a configurator (*e.g.* 210) that loads configuration information for use by the controller from a configuration file (*e.g.* 212) and stores the configuration information for subsequent access." *See, e.g., id.* at paragraphs 30, 36 and 37; *see also* FIGS. 2 and 3. The operation of the claimed controller and configurator is explained in the specification, as follows:

> **[0036]** A web server 202 hosts a web application 204, which may be accessed by using a browser 206 or the like. The web application 204 embodies a controller 208, which may be created according to the WPA controller architecture 102 (FIG. 2). Also included in the web application 204 is a configurator 210, which is constructed according to the configuration manager architecture 116 (FIG. 2). Although only a single configurator 210 is illustrated in FIG. 3, multiple configurators may be implemented and each may be responsible for loading

> configuration information related to a specific functionality. Those of ordinary skill in the art will appreciate that the configurator 210 (or configurators) may comprise one or more operational modules and may be separate from or integral with the web application 204. Additionally, the controller 208 may be integral with the web application 204 or it may function as a separate operational module.
>
> **[0037]** The configurator 210 may have the purpose of providing a centralized way of loading startup objects that may be required during the operation of the web application 204. For example, the configurator 210 may be adapted to load data from the configuration file or files 212 upon execution of the init() method of the controller 208. Data loaded by the configurator 210 may be stored in one or more configuration files 212, which may comprise one or more text properties configuration files or the like. After data from the configuration file or files 212 is loaded by the configurator 210, the data may be stored as a singleton object 214. A singleton object is an object that exists in memory such that only one of that type of object exists at any time in memory. Once created, a singleton object is not destroyed after use, like most objects, but is kept in memory until accessed again.
>
> Specification, paragraphs 36-37.

Independent claim 8 provides "a method of creating web applications." The method comprises "creating, with a processor-based device (*e.g.*, 102), a controller (*e.g.*, 18, 208) that receives a request (*e.g.*, 148) for data from a user (*e.g.*, 14) and responds to the request by sending information to the user...." *See, e.g., id.* at paragraphs 10, 15, 18, 19, 21, 22 and 36; *see also* FIGS. 1-3. Claim 8 additionally recites "providing a configurator (*e.g.*, 210) that loads configuration information for use by the controller from a configuration file (*e.g.*, 212) and stores the configuration information for subsequent access." *See, e.g., id.* at paragraphs 30, 36 and 37; *see also* FIGS. 2 and 3. The operation of the claimed controller and configurator is explained in the specification, as follows:

> **[0036]** A web server 202 hosts a web application 204, which may be accessed by using a browser 206 or the like. The web application 204 embodies a controller 208, which may be created according to the WPA controller architecture 102 (FIG. 2). Also included in the web application 204 is a configurator 210, which is constructed according to the configuration manager architecture 116 (FIG. 2). Although only a single configurator 210 is illustrated in FIG. 3, multiple configurators may be implemented and each may be responsible for loading configuration information related to a specific functionality. Those of ordinary skill in the art will appreciate that the configurator 210 (or configurators) may comprise one or more operational modules and may be separate from or integral with the web application 204. Additionally, the controller 208 may be integral with the web application 204 or it may function as a separate operational module.
>
> **[0037]** The configurator 210 may have the purpose of providing a centralized way of loading startup objects that may be required during the operation of the web application 204. For example, the configurator 210 may be adapted to load data from the configuration file or files 212 upon execution of the init() method of the controller 208. Data loaded by the configurator 210 may be stored in one or more configuration files 212, which may comprise one or more text properties configuration files or the like. After data from the configuration file or files 212 is loaded by the configurator 210, the data may be stored as a singleton object 214. A singleton object is an object that exists in memory such that only one of that type of object exists at any time in memory. Once created, a singleton object is not destroyed after use, like most objects, but is kept in memory until accessed again.
>
> Specification, paragraphs 36-37.

Independent claim 15 provides "a system for creating web applications." The system comprises "means (*e.g.*, 102) for creating a controller (18, 208) that is adapted to receive a request (*e.g.*, 148) for data from a user (*e.g.*, 14) and respond to the request..." *See, e.g., id.* at paragraphs 10, 15, 18, 19, 21, 22 and 36; *see also* FIGS. 1-3. Claim 15 further recites "means (*e.g.*, 116) for creating a configurator (*e.g.*, 210) that loads configuration information for use by the controller from a configuration file (*e.g.*, 212) and stores the configuration information for subsequent access." *See, e.g., id.* at paragraphs 18, 30, 36 and 37; *see also* FIG. 2. The operation of the claimed controller and configurator is explained in the specification, as follows:

> **[0036]** A web server 202 hosts a web application 204, which may be accessed by using a browser 206 or the like. The web application 204 embodies a controller 208, which may be created according to the WPA controller architecture 102 (FIG. 2). Also included in the web application 204 is a configurator 210, which is constructed according to the configuration manager architecture 116 (FIG. 2). Although only a single configurator 210 is illustrated in FIG. 3, multiple configurators may be implemented and each may be responsible for loading configuration information related to a specific functionality. Those of ordinary skill in the art will appreciate that the configurator 210 (or configurators) may comprise one or more operational modules and may be separate from or integral with the web application 204. Additionally, the controller 208 may be integral with the web application 204 or it may function as a separate operational module.
>
> **[0037]** The configurator 210 may have the purpose of providing a centralized way of loading startup objects that may be required during the operation of the web application 204. For example, the configurator 210 may be adapted to load data from the configuration file or files 212 upon

> execution of the init() method of the controller 208. Data loaded by the configurator 210 may be stored in one or more configuration files 212, which may comprise one or more text properties configuration files or the like. After data from the configuration file or files 212 is loaded by the configurator 210, the data may be stored as a singleton object 214. A singleton object is an object that exists in memory such that only one of that type of object exists at any time in memory. Once created, a singleton object is not destroyed after use, like most objects, but is kept in memory until accessed again.
>
> Specification, paragraphs 36-37.

Independent claim 22 provides "a machine readable medium" (*e.g.*, a structure readable by a processor-based device) that stores computer-readable instructions. *See, e.g., id.* at paragraphs 17. In particular, the machine readable medium comprises "a controller generator (*e.g.*, 102) stored on the machine readable medium, the controller generator being adapted to provide a web application (*e.g.*, 204) with a controller (*e.g.*, 18, 208) that receives a request (*e.g.*, 148) for data from a user (*e.g.*, 14) and responds to the request by sending information to the user...." *See, e.g., id.* at paragraphs 10, 15, 18, 19, 21, 22 and 36; *see also* FIGS. 1-3. Claim 22 further recites "a configurator generator stored on the machine readable medium, the configurator generator being adapted to provide a configurator (*e.g.*, 210) that loads configuration information for use by the controller from a configuration file (*e.g.*, 212) and stores the configuration information for subsequent access. *See, e.g., id.* at paragraphs 18, 30, 36 and 37; *see also* FIG. 2. The operation of the claimed controller and configurator is explained in the specification, as follows:

[0036] A web server 202 hosts a web application 204, which may be accessed by using a browser 206 or the like. The web application 204 embodies a controller 208, which may be created according to the WPA controller architecture 102 (FIG. 2). Also included in the web application 204 is a configurator 210, which is constructed according to the configuration manager architecture 116 (FIG. 2). Although only a single configurator 210 is illustrated in FIG. 3, multiple configurators may be implemented and each may be responsible for loading configuration information related to a specific functionality. Those of ordinary skill in the art will appreciate that the configurator 210 (or configurators) may comprise one or more operational modules and may be separate from or integral with the web application 204. Additionally, the controller 208 may be integral with the web application 204 or it may function as a separate operational module.

[0037] The configurator 210 may have the purpose of providing a centralized way of loading startup objects that may be required during the operation of the web application 204. For example, the configurator 210 may be adapted to load data from the configuration file or files 212 upon execution of the init() method of the controller 208. Data loaded by the configurator 210 may be stored in one or more configuration files 212, which may comprise one or more text properties configuration files or the like. After data from the configuration file or files 212 is loaded by the configurator 210, the data may be stored as a singleton object 214. A singleton object is an object that exists in memory such that only one of that type of object exists at any time in memory. Once created, a singleton object is not destroyed after use, like most objects, but is kept in memory until accessed again.

Specification, paragraphs 36-37.

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

First Ground of Rejection for Review on Appeal:

Appellants respectfully urge the Board to review and reverse the Examiner's first ground of rejection in which the Examiner rejected claims 1-7 and 15-21 under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

Second Ground of Rejection for Review on Appeal:

Appellants respectfully urge the Board to review and reverse the Examiner's second ground of rejection in which the Examiner rejected claims 1-24 under 35 U.S.C. § 102(a) as anticipated by Kwong et al. (Building a Portlet within the Model-View Controller Paradigm Using Web Sphere® Portal, hereinafter "the Kwong reference").

7. ARGUMENT

As discussed in detail below, the Examiner has improperly rejected the pending claims. Further, the Examiner has misapplied long-standing and binding legal precedents and principles in rejecting the claims under Sections 101 and 102. Accordingly, Appellants respectfully request the Board to reverse all pending rejections.

A. First Ground of Rejection:

With respect to the rejection of claims 1-7 and 15-21, the Examiner's rejection of independent claims 1 and 15 is exemplary:

> Claims 1 - 7 and 15 - 21 (see Examiner's Note above) are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claims 1 and 15.
>
> As disclosed in the specification of the application, all components recited in the claim that constitute the claimed system are constructed of software program objects and/or instructions (Figs 1 - 4 and associated text). Thus, the system is considered as software programs system claimed as computer listings, per se, i.e. containing machine-executable instructions; therefore, it is non-statutory according to 35 U.S.C 101. See MPEP 2106.01 (I): "...the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed ... ". For the purpose of further claim analysis under 35 U.S.C. 102 and 103, The Examiner treats both Claims 1 and 15 as a computer program containing machine-readable instructions stored on a physical medium for performing the method or steps recited in the claim.
>
> Final Office Action, pp. 12-13.

The Appellants respectfully assert that the present claims are directed to statutory subject matter. Any analysis of whether a claim is directed to statutory subject matter begins with the language of 35 U.S.C. § 101, which reads:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore, subject to the conditions and requirements of this title.

In interpreting Section 101, the Supreme Court stated that Congress intended statutory subject matter to "include *anything* under the sun that is made by man." *Diamond v. Chakrabarty*, 447 U.S. 303, 309, 206 U.S.P.Q. 193, 197 (1980) (emphasis added). Although this statement may appear limitless, the Supreme Court has identified three categories of unpatentable subject matter: laws of nature, natural phenomena, and abstract ideas. *See*, *Diamond v. Diehr*, 450 U.S. 175, 182, 209 U.S.P.Q. 1, 7 (1981). Accordingly, so long as a claim is not directed to one of the three specific areas listed above, the claim is directed to patentable subject matter. Thus, it is improper to read restrictions into Section 101 regarding subject matter that may be patented where the legislative history does not indicate that Congress clearly intended such limitation. *In re Alappat*, 31 U.S.P.Q.2d 1545, 1556 (Fed. Cir. 1994) (citing *Chakrabarty* 447 U.S. at 308).

For example, the fact that a claim includes or is directed to an algorithm is no ground for holding a claim is directed to non-statutory subject matter. *See*, *In re Iwashashi*, 12 U.S.P.Q.2d 1908, 1911 (Fed. Cir 1989). Rather, the proscription against patenting an algorithm, to the extent it still exists, is narrowly limited to *mathematical algorithms in the abstract*, e.g., describing a mathematical algorithm as a procedure for solving a given type of mathematical problem. *See*, *AT&T Corp. v. Excel Communications, Inc.*, 50 U.S.P.Q.2d 1447, 1450 (Fed. Cir 1999). Indeed, the courts are aware that any step-by-step process, be it electronic, chemical, or mechanical, involves an algorithm. *Id.* at 1450.

Thus, inquiry into what is statutory subject matter simply requires "an examination of the contested claims to see if the claimed subject matter as a whole is a disembodied mathematical concept representing nothing more than a 'law of nature' or an 'abstract idea, or if the mathematical concept has been reduced to some practical application rendering it 'useful'" *Id.* at 1451 (citing and quoting *In re Alappat*, 31 U.S.P.Q.2d at 1557). Furthermore, a Section 101 analysis "demands that the focus in any statutory subject matter analysis be on the *claim as a whole*." *In re Alappat*, 31 U.S.P.Q.2d at 1557 (citing *Diehr*, 450 U.S. at 192) (emphasis in original). Indeed, the dispositive inquiry is whether the claim *as a whole* is directed to statutory subject matter, it is irrelevant that a claim may contain, as part of the whole, subject matter that would not be patentable by itself. *Id.*

The Appellants respectfully disagree with the Examiner's assertions and interpretation of the law. Indeed, the Appellants contend that the Examiner's assertion that software inventions are *per se* non-statutory flies in the face of the clear precedent of the Federal Circuit, as set forth above. Moreover, the systems recited in independent claims 1 and 15 are clearly useful for "creating web applications" and are fully supported by the specification as set forth above. This is all the law requires in order to comply with Section 101. Accordingly, Appellants submit that independent claims 1 and 15 are directed to statutory subject matter.

In addition, the Examiner's reliance on *In re Warmerdam* as set forth in M.P.E.P. Section 2106.01 is misplaced. In *Warmerdam*, the Federal Circuit concluded that the applicant's method claims were directed to nonstatutory subject matter because they related to *method steps* involving the mere manipulation of abstract ideas. With respect to the applicant's claim directed to "a machine," the court stated that those claims were "clearly patentable subject matter." *Warmerdam*, 31 U.S.P.Q. 2d at 1759. Appellants' claims in this case, which are directed to "a system," are clearly statutory in view of *Warmerdam*.

In view of the arguments presented above, Appellants contend that the rejection of claims 1-7 and 15-21 is improper. Accordingly, Appellants request the Board to reverse the Examiner's rejection of claims 1-7 and 15-21 under Section 101.

B. **Second Ground of Rejection:**

With respect to the rejection of claims 1-24, the Examiner's rejections of independent claims 1, 8, 15 and 22 are exemplary:

> -- Claims 1, 8, 15, and 22: Kwong et al. disclose a software program tangibly stored on a machine readable medium containing computer readable instructions for performing the method of creating web applications, the method comprising:
>
> creating a controller that receives a request for data from a user and responds to the request by sending information to the user (i.e., a portal and its inherent properties; Page 1); and providing a configurator that loads configuration information for use by the controller from a configuration file and stores the configuration information for subsequent access (i.e., a portlet and its specified deployment and user attributes; Page 2).
>
> Final Office Action, pp. 13, 14.

In rejecting the claims, the Examiner further asserted that:

> A portal, as well understood by one of ordinary skill in the art, is capable of offering a broad array of resources and services, such as e-mail, forums, search engines, etc. Thus, a portal is a controller generator (Examiner's explanation: a portal can contain different types of portlets and the portlets of a portal can have different configurations.
>
> Final Office Action, p. 7.

1. **Judicial precedent has clearly established a legal standard for a *prima facie* anticipation rejection.**

Anticipation under Section 102 can be found only if a single reference shows exactly what is claimed. *Titanium Metals Corp. v. Banner*, 227 U.S.P.Q. 773 (Fed. Cir. 1985). Thus, for a prior art reference to anticipate under Section 102, every element of the claimed invention must be identically shown in a single reference. *In re Bond*, 15 U.S.P.Q.2d 1566 (Fed. Cir. 1990). Moreover, the prior art reference also must show the *identical* invention "*in as complete detail as contained in the ... claim*" to support a *prima facie* case of anticipation. *Richardson v. Suzuki Motor Co.*, 9 U.S.P.Q. 2d 1913, 1920 (Fed. Cir. 1989) (emphasis added). Accordingly, Appellants need only point to a single element not found in the cited reference to demonstrate that the cited reference fails to anticipate the claimed subject matter.

2. **The Examiner's rejection of independent claims 1, 8, 15 and 22 is improper because the rejection fails to establish a *prima facie* case of anticipation.**

Independent claim 1 recites:

> A system for creating web applications, the system comprising:

> a controller generator that is *adapted to provide a web application with a controller* that receives a request for data from a user and responds to the request by sending information to the user; and
> a configurator generator that is adapted *to provide a configurator* that loads *configuration information for use by the controller* from a configuration file and stores the configuration information for subsequent access.
>
> (Emphasis added.)

Independent claim 8 recites:

> A method of creating web applications, the method comprising:
> creating, with a processor-based device, *a controller* that receives a request for data from a user and responds to the request by sending information to the user; and
> providing a configurator that loads *configuration information for use by the controller* from a configuration file and stores the configuration information for subsequent access.
>
> (Emphasis added.)

Independent claim 15 recites:

> A system for creating web applications, the system comprising:
> means for creating a *controller* that is adapted to receive a request for data from a user and respond to the request; and
> means for creating a *configurator that loads configuration information for use by the controller* from a configuration file and stores the configuration information for subsequent access.
>
> (Emphasis added.)

Independent claim 22 recites:

> A machine readable medium, comprising:
> a controller generator stored on the machine readable medium, the controller generator being adapted *to provide a web application with a controller* that receives a request for data from a user and responds to the request by sending information to the user; and
> a configurator generator stored on the machine readable medium, the configurator generator being adapted to provide *a configurator that loads configuration information for use by the controller* from a configuration file and stores the configuration information for subsequent access.
>
> (Emphasis added.)

Appellants respectfully submit that the rejection of independent claims 1, 8, 15 and 22 is improper because the prior art reference that is used to reject the claims does not disclose each and every element recited in those claims. Specifically, it appears that the Examiner interpreted the claimed web application and controller as a single entity and, further so, equated such an entity to a portal disclosed by the Kwong reference. First, Appellants respectfully note that the controller and the web application recited by the claims are two distinct separate entities. As stated in the specification:

> **[0015]** The controller 18 functions as an intermediary between the client 14 and the model object 12 and view 16 of the application. For example, the controller 18 can manage access by the view 16 to the model 12 and, also, manage notifications and changes of data among objects of the view 16 and objects of the model 12. The control and flow logic 24 of the controller 18 also may be subdivided into model-controllers and view-controllers to address and respond to various control issues of the model 12 and the view 16, respectively. Accordingly, the model-controllers manage the models 12 and communicate with view-controllers, while the view-controllers manage the views 16 and communicate with the model-controllers. Subdivided

> or not, the controllers 18 ensure communication and consistency between the model 12, the view 16 and the client 14.
>
> ***
>
> **[0036]** The web application 204 embodies a controller 208, which may be created according to the WPA controller architecture 102 (FIG. 2). Also included in the web application 204 is a configurator 210, which is constructed according to the configuration manager architecture 116 (FIG. 2). Although only a single configurator 210 is illustrated in FIG. 3, multiple configurators may be implemented and each may be responsible for loading configuration information related to a specific functionality. Those of ordinary skill in the art will appreciate that the configurator 210 (or configurators) may comprise one or more operational modules and may be separate from or integral with the web application 204. Additionally, the controller 208 may be integral with the web application 204 or it may function as a separate operational module.
>
> Specification, paragraphs 15, 36.

Thus, the controller and the model objects of the web application operate distinctly such that both elements cannot be characterized as a single component in a model view controller (MVC) application. Furthermore, there is no disclosure in the Kwong reference that would indicate that a portal can support the distinctive operations and unique functionalities of the claimed controller and web application.

The Examiner has equated the portals described in the Kwong reference to the claimed controller. This comparison is not correct. Appellants respectfully note that a portal may be described as:

> A web site that serves as a gateway to the Internet. A portal is a collection of links, content, and services designed to guide users to information they are likely to find interesting-news, weather, entertainment, commerce sites, chat rooms, and so on.
>
> Microsoft Computer Dictionary, Fifth Edition, page 413 (copy attached as Exhibit 1).

In contrast, the claimed controller is part of web presentation architecture WPA having aspects such as a preprocessor, a localization manager, a navigation manager, a layout manager, a cookie manager, an object cache manager and/or a configuration manager. Specification, paragraph 18. The controller is also responsible for "various control and flow logic among the various model-view-controller divisions of the WPA 100." Specification, paragraph 19. The Kwong reference teaches designing portlets within portals, such as a WebSphere portal, which according to the Kwong reference:

> lets you build portals by combining software components known as portlets. Developers can build portlets independently of one another, and reuse and combine them to build different portals.
>
> Kwong, page 1, introduction section.

Yet, there are no teachings in the Kwong reference that indicate that a portlet and/or portal posses any of the above-mentioned components and/or properties of the claimed controller. Accordingly, the Kwong reference clearly does not teach a controller generator that is adapted to provide a web application with a controller that receives requests for data, as recited by independent claims 1, 8, 15 and 21.

Further, because the Kwong reference does not teach the claimed controller, Kwong cannot disclose a system having a configurator that loads configuration information for use by the *controller*, as recited by independent claims 1, 8, 15 and 22.

For at least these reasons, the Examiner has not established a *prima facie* case of anticipation with respect to independent claims 1, 8, 15 and 22. Accordingly, Appellants respectfully request the Board to reverse the rejection under Section 102 of independent claims 1, 8, 15 and 22, as well as those claims depending there from.[1]

Kwong Does Not Anticipate dependent Claims 3, 10, 17 and 23

The rejection of dependent claims 3, 10, 17 and 23 is improper because the Kwong reference does not disclose each and every element recited by the claims. For example, dependent claims 3, 10, 17 and 23 recite a configurator adapted "to store the *configuration information as a singleton object*." (emphasis added). As appreciated by those of ordinary skill in the art in light of Appellants' disclosure, a singleton object is:

> an object that exists in memory such that only one of that type of object exists at any time in memory. Once created, a singleton object is not destroyed after use, like most objects, but is kept in memory until accessed again.
>
> Specification, paragraph 37.

[1] In the Final Office Action, the Examiner referenced additional documents ("WebSphere Portal: Portlet Concepts," and Guidelines, and "Introducing the Portlet Specification.") as providing support for the rejection under Section 102. These documents have not been established as prior art under 35 U.S.C. § 102. Accordingly, Appellants have not addressed those references in this Appeal Brief.

The Kwong reference does not disclose any singleton object at all. In particular, no singleton object having the properties described above is disclosed in Kwong. Therefore, the Kwong reference cannot anticipate the subject matter of claims 3, 10, 17 and 23.

In rejecting claims 3, 10, 17 and 23, the Examiner interpreted the recited singleton object as a specific portlet. Final Office Action, page 14. However, in the rejection of independent claims 1, 8, 15 and 22, the Examiner interpreted the portlet of the Kwong reference as the configurator generator. Appellants submit that, in the context of dependent claims 3, 10, 17 and 23, a configurator generator provides a configurator that loads data from a configuration file that is stored as a singleton object. Specification, paragraph 37. Thus, a configurator generator and a singleton object are two distinct entities. In the Final Office Action, the Examiner set forth that the "the Examiner could also refer to the portlet itself as a singleton object since it also meets Applicant's definition." Final Office Action, page 10. Yet, the Examiner has not pointed to any portions of the Kwong reference equating the claimed singleton object to Kwong's portlet. Appellants respectfully submit that merely asserting such equivalence is insufficient for establishing a *prima facie* case of obviousness. Based on the foregoing discussion, the Examiner's contention that the portlet of the Kwong reference corresponds to the claimed configurator generator is internally inconsistent with the rejection of the independent claims. For at least these reasons, the rejection of claims 3, 10, 17 and 23 is defective. Appellants request the Board to reverse the Examiner's rejection and allow dependent claims 3, 10, 17 and 23.

Conclusion

Appellants respectfully submit that all pending claims are in condition for allowance. However, if the Board wishes to resolve any issues by way of a telephone conference, the Board is kindly invited to contact the undersigned attorney at the telephone number indicated below.

Respectfully submitted,

Date: May 23, 2007

Barry D. Blount

Barry D. Blount
Reg. No. 35,069
FLETCHER YODER
(281) 970-4545

CORRESPONDENCE ADDRESS:
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

8. APPENDIX OF CLAIMS ON APPEAL

Listing of Claims:

1. A system for creating web applications, the system comprising:

a controller generator that is adapted to provide a web application with a controller that receives a request for data from a user and responds to the request by sending information to the user; and

a configurator generator that is adapted to provide a configurator that loads configuration information for use by the controller from a configuration file and stores the configuration information for subsequent access.

2. The system set forth in claim 1, wherein the configuration file is a text properties configuration file.

3. The system set forth in claim 1, wherein the configurator is adapted to store the configuration information as a singleton object.

4. The system set forth in claim 1, wherein the configuration information comprises error handling information.

5. The system set forth in claim 1, wherein the configuration information comprises log processing information.

6. The system set forth in claim 1, wherein the configuration information comprises data that is specific to each of a plurality of portals.

7. The system set forth in claim 1, wherein the configurator is adapted to read the configuration information upon initialization of the controller.

8. A method of creating web applications, the method comprising:

creating, with a processor-based device, a controller that receives a request for data from a user and responds to the request by sending information to the user; and

providing a configurator that loads configuration information for use by the controller from a configuration file and stores the configuration information for subsequent access.

9. The method set forth in claim 8, comprising defining the configuration file to be a text properties configuration file.

10. The method set forth in claim 8, comprising adapting the configurator to store the configuration information as a singleton object.

11. The method set forth in claim 8, comprising defining the configuration information to comprise error handling information.

12. The method set forth in claim 8, comprising defining the configuration information to comprise log processing information.

13. The method set forth in claim 8, comprising defining the configuration information to comprise data that is specific to each of a plurality of portals.

14. The method set forth in claim 8, comprising adapting the configurator to read the configuration information upon initialization of the controller.

15. A system for creating web applications, the system comprising:

means for creating a controller that is adapted to receive a request for data from a user and respond to the request; and

means for creating a configurator that loads configuration information for use by the controller from a configuration file and stores the configuration information for subsequent access.

16. The system set forth in claim 15, wherein the configuration file is a text properties configuration file.

17. The system set forth in claim 15, wherein the configurator is adapted to store the configuration information as a singleton object.

18. The system set forth in claim 15, wherein the configuration information comprises error handling information.

19. The system set forth in claim 15, wherein the configuration information comprises log processing information.

20. The system set forth in claim 15, wherein the configuration information comprises data that is specific to each of a plurality of portals.

21. The system set forth in claim 15, wherein the configurator is adapted to read the configuration information upon initialization of the controller.

22. A machine readable medium, comprising:

a controller generator stored on the machine readable medium, the controller generator being adapted to provide a web application with a controller that receives a request for data from a user and responds to the request by sending information to the user; and

a configurator generator stored on the machine readable medium, the configurator generator being adapted to provide a configurator that loads configuration information for use by the controller from a configuration file and stores the configuration information for subsequent access.

23. The machine readable medium set forth in claim 22, wherein the configurator generator is adapted to produce a configurator that stores the configuration information as a singleton object.

24. The machine readable medium set forth in claim 22, wherein the configurator generator is adapted to produce a configurator that reads the configuration information upon initialization of the controller.

9. <u>EVIDENCE APPENDIX</u>

None.

10. **RELATED PROCEEDINGS APPENDIX**

None.